Lesson 1 - MATLAB Basics

As has been said countless times by Cooper students, MATLAB can be frustrating to use sometimes. Even so, MATLAB is a pretty flexible language with a lot of neat features.

Note: Before reading the rest of this, make sure that you have created a MathWorks account using your **cooper.edu email** as the school has a license for us to use MATLAB.

Once you have created an account with your cooper.edu email, you can either access MATLAB through your browser on the MathWorks website or download MATLAB as an application on your computer. When you download MATLAB, be sure to include the Signal Processing Toolbox when including packages to install.

The MATLAB environment

Upon starting up MATLAB and opening up a new script, the application should look something like the picture below.

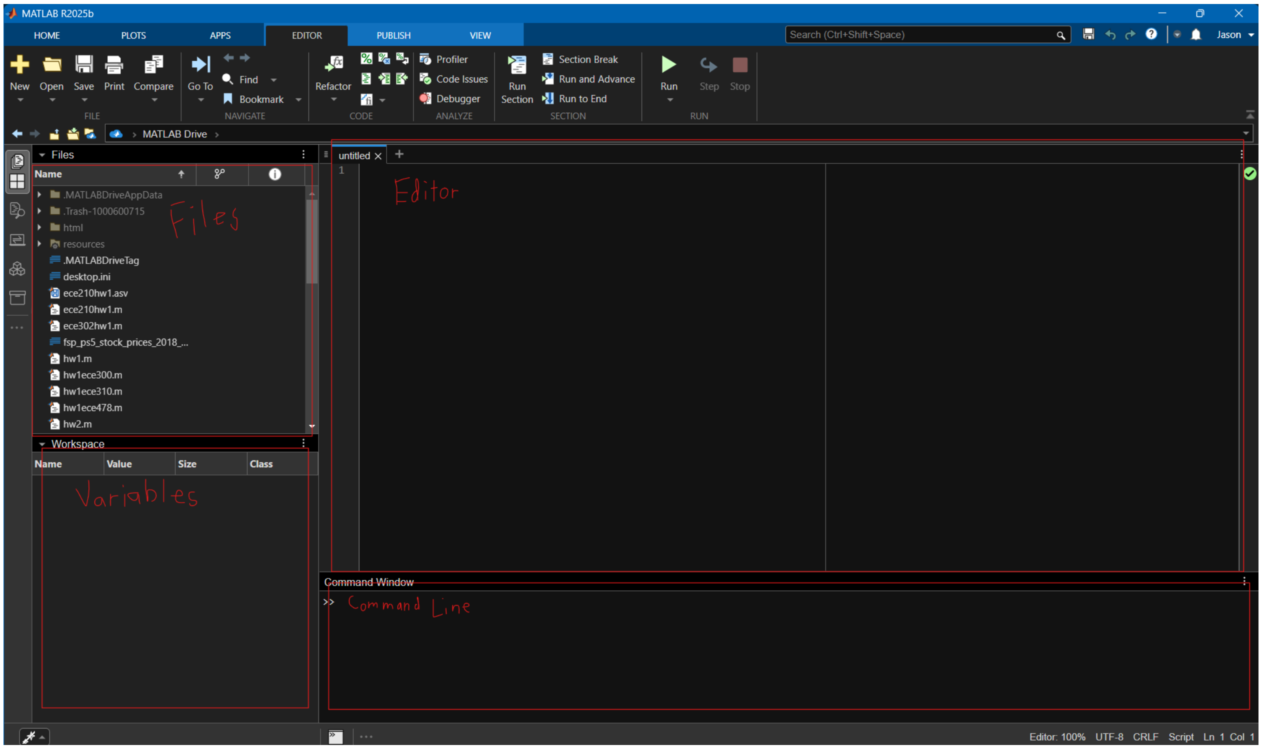

Figure 1: The MATLAB GUI

The editor is where you type and edit your MATLAB scripts for homeworks and other projects. Whenever you want to run/save a script, you can click the run button at the very top to execute the script. At the very bottom is the command window where you can directly execute commands and run programs in MATLAB. Analogous to Python, the command line is like the Python interactive shell where commands that are entered are immediately executed.

The bottom left shows all of the values of the variables defined after a script is run. Similar to Google Colab, the values of variables are saved between running scripts, so keep this in mind. All of the files in the working directory of your MATLAB Drive are also shown on the sides.

Fundamentals

Unsuprisingly, the vast majority of variables you will be dealing with in MATLAB are matrices. Most variables you work with will be in the form of a scalar (1x1 matrix), row/column vector (1xn / nx1 matrices), or 2 dimensional matrices (mxn matrices). However, it is also possible for variables to have more than 2 dimensions. Here are some ways to instantiate variables and matrices in MATLAB.

```matlab
% Note: Semicolons mute the output to the command line.
% Percent signs also allow us to make comments.

scalar = 3;

row = [1, 2, 3, 4]; % Making a 1x4 row vector
column = [1i; 3i; 5i; 7i]; % Making a 4x1 column vector of complex numbers

A = [5, 6; 7, 8] % 2x2 matrix
a = [
        4, pi
        -3, 2
    ]; % You can also make matrices like this.
% MATLAB is case sensitive

B = [A, a]; % We can concatenate matrices together
```

Some Operations

Here are some basic array and matrix operators that are available in MATLAB. Array operations act on each element individually while matrix operations treat the variable as a matrix.

Note: The matrix transpose operator takes the complex conjugate of all entries when transposing.

MATLAB also has some additional mathematical operators using built-in functions like sqrt() for square root and exp() for exponentiation by e.

Operator	Purpose
A + B	Addition
A - B	Subtraction
A .* B	Elementwise Multiplication
A ./ B	Array Right Division
A .\ B	Array Left Division
A .ˆB	Elementwise Exponentiation
A .’	Array Transpose

Table 1: Array Operations

Operator	Purpose
A * B	Matrix Multiplication
A / B	Matrix Right Division
A \ B	Matrix Left Division
A ’	Matrix (Hermitian) Transpose

Table 2: Matrix Operations

Documentation

There are a lot more features and functions available in MATLAB that I can possibly cover in this course. If you don’t know what some function does or what kinds of inputs it accepts, you can use the help and doc commands, You can also consult the offical MATLAB documentation at `www.mathworks.com/help/matlab` or `www.mathworks.com/help/pdf_doc/matlab`.